

# Aseguramiento de la Calidad del Software:Avance

1

Instituto Tecnológico de Costa Rica,Escuela de Ingeniería en Computación

October 23, 2016

**Hecho por: Ricardo Castro 201231830**

**Andrés Jimenez 2013015630**

**Jean Carlo Mata 2013011325**

# Contents

<b>I</b>	<b>(SyRS)</b>	<b>5</b>
<b>1</b>	<b>Scope</b>	<b>5</b>
<b>2</b>	<b>Conformance.</b>	<b>5</b>
2.1	Intended Usage: . . . . .	5
2.2	Conformance to processes: . . . . .	6
<b>3</b>	<b>Normative references</b>	<b>6</b>
<b>4</b>	<b>Terms, definitions and abbreviated terms</b>	<b>6</b>
4.1	Terms and definitionsn . . . . .	6
4.1.1	Acquirer . . . . .	6
4.1.2	Attribute . . . . .	6
4.1.3	Baseline: . . . . .	6
4.2	Abbreviated terms . . . . .	7
<b>5</b>	<b>Concepts</b>	<b>7</b>
5.1	Introduction . . . . .	7
5.2	Requirements fundamentals . . . . .	7
5.2.1	General . . . . .	7
5.2.2	Stakeholders . . . . .	7
5.2.3	Transformation of needs into requirements . . . . .	7
5.2.4	Requirements construct . . . . .	8
5.2.5	Characteristics of individual requirements . . . . .	8
5.2.6	Characteristics of a set of requirements . . . . .	8
5.2.7	Requirement language criteria . . . . .	8
5.2.8	Requirements attributes . . . . .	9
5.3	Practical considerations . . . . .	10
5.3.1	Iteration and recursion of processes . . . . .	10
5.4	Requirement information items . . . . .	10
<b>6</b>	<b>Procceses</b>	<b>11</b>
6.1	Requirement processes . . . . .	11
6.1.1	Guidelines for Processes . . . . .	11
6.2	Stakeholder requirements definition process . . . . .	12
6.2.1	Purpose . . . . .	12
6.2.2	Outcomes . . . . .	12
6.2.3	Activities and tasks . . . . .	12
6.3	Requirements analysis process . . . . .	12
6.3.1	Purpose . . . . .	12
6.3.2	Outcomes . . . . .	13
6.3.3	Activities and tasks . . . . .	13
6.4	Requirements engineering activities in other technical processes .	13

6.4.1	Requirements in architectural design . . . . .	13
6.4.2	Requirements in verification . . . . .	13
6.4.3	Requirements in validation . . . . .	13
6.5	Requirements management . . . . .	13
6.5.1	Management Overview . . . . .	13
6.5.2	Change management . . . . .	14
6.5.3	Measurement for requirements . . . . .	14
<b>7</b>	<b>Information items</b>	<b>14</b>
<b>8</b>	<b>Guidelines for information items</b>	<b>14</b>
8.1	Requirements information item outlines . . . . .	14
8.2	Stakeholder requirements specification document . . . . .	14
8.3	System requirements specification document . . . . .	14
<b>9</b>	<b>Information item content</b>	<b>15</b>
9.1	Introduction . . . . .	15
9.2	General content . . . . .	15
9.3	Stakeholder requirements specification (StRS) document. . . . .	15
9.4	System requirements specification (SyRS) document . . . . .	15
9.4.1	System purpose . . . . .	15
9.4.2	System scope . . . . .	15
9.4.3	System overview . . . . .	15
9.4.4	Functional requirements . . . . .	16
9.4.5	Usability requirements . . . . .	17
9.4.6	Performance requirements . . . . .	17
9.4.7	System interfaces . . . . .	18
9.4.8	System Operations . . . . .	18
9.4.9	System modes and states If the system can exist in various modes or states define these and, as appropriate, use diagrams. . . . .	18
9.4.10	Physical characteristics . . . . .	18
9.4.11	Environmental conditions . . . . .	19
9.4.12	System security . . . . .	19
9.4.13	Information management . . . . .	19
9.4.14	Policies and regulations . . . . .	19
9.4.15	System life cycle sustainment . . . . .	19
9.4.16	Packaging, handling, shipping and transportation . . . . .	19
9.4.17	Verification . . . . .	20
9.4.18	Assumptions and dependencies . . . . .	20

## II Estandar de codicacion a utilizar a lo largo del proyecto semestral 20



# Part I

## (SyRS)

### 1 Scope

#### a) Nombre: Clasificador de Jugadores

El alcance del presente proyecto consiste en realizar un sistema que permita a los a los dirigentes del Club Atlético Boca Juniors realizar un complejo estudio sobre los datos de sus jugadores por lineas en el campo(defensa,medio campo y delantera) , para lograr esto se hace un procesamiento de vídeos, en el cual se fragmenta el vídeo, se analiza imagen por imagen y luego se procesan estas imágenes, separando los jugadores de ambos equipos, se aplica un análisis automatizado para obtener las estrategias y tácticas de cada uno de los equipos.

Se podran ver los cambios en tiempo real asi como descargar el video una vez finalizado el proceso.

El proyecto se compone de varias etapas, las cuales son: la identificación de escenas con información útil en el vídeo, la segmentación y rastreo de jugadores y análisis automático de recorridos y comportamiento de jugadores.

Para lograr el análisis de jugadores por medio del vídeo, el mismo se descompone y se analiza por colores para lograr separar los jugadores del equipo A y los jugadores del equipo B. Para lograr este objetivo se algoritmo de clasificación conocido como K-medias, con esto se llega a tener una clasificación de un conjunto de datos K sin necesidad de entrenamiento previo.

Este proyecto se realizará en java utilizando eclipse, opencv y su interface será en web utilizando angular para su implementación

### 2 Conformance.

#### 2.1 Intended Usage:

Este proyecto se pretende concretar para solventar una necesidad del Club Atlético Boca Juniors, el proyecto seria utilizado tanto por el presidente del club como por el resto de la junta directiva del mismo. Con el uso de este proyecto los directivos de boca pretenden

conocer el funcionamiento táctico del equipo en cada una de sus líneas(defensa,medio campo y delantera), para de esta manera corregir y mejorar el funcionamiento del equipo.

## 2.2 Conformace to processes:

El proceso para la realización del proyecto es la siguiente: desarrollo de la etapa de segmentación de jugadores y la clasificación de las regiones en la imagen correspondientes a los jugadores o «blobs» por color, de manera no supervisada, para determinar la pertenencia de cada jugador a alguno de los equipos. El algoritmo de clasificación no supervisada a implementar es el conocido K-medias , muy utilizado en diversas aplicaciones donde se requiere clasificar o amontonar un conjunto de datos en K clases, sin que sea necesaria una etapa de entrenamiento previa.

## 3 Normative references

ISO/IEC/IEEE 29148:2011 Systems and software engineering — Life cycle processes — Requirements engineering

## 4 Terms, definitions and abbreviated terms

### 4.1 Terms and definitions

#### 4.1.1 Acquirer

La junta directiva del Club Atlético boca juniors de Argentina.

#### 4.1.2 Attribute

El precio del proyecto esta tasado en 3millones de pesos argentinos . Este producto no será comercializado, será únicamente para los directivos de boca.

#### 4.1.3 Baseline:

El producto es un software para procesamiento de vídeos, brindado a los dirigentes del Club Atlético Boca Juniors para que ellos puedan hacer un análisis táctico de las líneas de juego. Esto se hace por medio de algoritmos de segmentación de jugadores y clasificación en diferentes regiones, se crean blobs por color para determinar las posiciones de los jugadores, además se implemente un algoritmo de clasificación conocido como el algoritmo k-medias .

## 4.2 Abbreviated terms

CABJ (Club Atlético Boca Juniors)

StRS (Stakeholder requirements specification document )

SyRS (System requirements specification document)

## 5 Concepts

### 5.1 Introduction

Para este proyecto se utilizaran varios conceptos pocos conocidos para el usuario, algunos de estos conceptos se citan a continuación:

**Segmentación y rastreo de jugadores:** Es el propósito más general del proyecto, para lograrlo se utilizan una serie de algoritmos que al final, separaran a los jugadores de ambos equipos del resto del terreno de juego y así se podrá observar el funcionamiento táctico del equipo linea por linea.

**Algoritmo K-medias:** es un algoritmo de clasificación que se utilizara para la segmentación y rastreo de jugadores.

### 5.2 Requirements fundamentals

#### 5.2.1 General

El principal requerimiento de los directivos de CABJ es obtener un análisis táctico por linea de sus jugadores, esto con el fin de mejorar el rendimiento del equipo para seguir la racha ganadora que siempre ha tenido el equipo.

#### 5.2.2 Stakeholders

Los stakeholders en este caso son las personas del alto mando de CABJ esto porque son las personas para las cuales se esta produciendo este software, además son los que operaran el sistema y los que se verán beneficiados por el mismo.

#### 5.2.3 Transformation of needs into requirements

El principal objetivo de la junta directiva de Boca es obtener títulos, ya que en este momento se cuenta con una sequía de 2 campeonatos cortos sin ganar el cetro, por esta situación el presidente del equipo esta intentando una mejora en el funcionamiento del mismo, para

este fin encargó un software a nuestra compañía(Sport Analytics) el desarrollo de un software que pueda procesar vídeos de tal forma que se entregue como resultado un análisis línea por línea del equipo, para saber el parado y acomodo del mismo y poder hacer cambios tácticos si el caso lo amerita.

#### 5.2.4 Requirements construct

Para poder cumplir con los deseos de los directivos de Boca la compañía Sport Analytics se compromete con los altos dirigentes de CABJ con el fin de entregar un software que cumpla con lo siguiente. El software procesará vídeos de fútbol, cuando se estén procesando se pondrá un mensaje de espera, que indique que el vídeo se está procesando. El software analizará dichos vídeos en busca de blops para descubrir el parado del equipo línea por línea en dichos vídeos, una vez hecho esto se guardará el tiempo de procesamiento de dichos blops para más adelante ser mostrados a un usuario. Cuando el análisis esté completo se generará un informe con las duraciones de las escenas procesadas. Una vez finalizado el análisis se generará un informe con las posiciones de cada jugador en un cuadro, de un equipo específico.

#### 5.2.5 Characteristics of individual requirements

Cuando el vídeo se esté procesando saldrá un señal, ya sea un símbolo de “espera” o la palabra procesando o cargando. El tiempo que debe entregarse debe venir en una tabla indicando el número de escena procesada y el tiempo que se duró haciendo el análisis. Se darán también en una tabla los resultados de los análisis de los jugadores línea por línea(defensa medio campo y delantera).

#### 5.2.6 Characteristics of a set of requirements

El requerimiento principal es obtener un análisis detallado de la táctica utilizada en cada una de las líneas del equipo(defensa, medio campo y delantera) mediante el uso de un software encargado de procesamiento de vídeos, el mismo devolverá junto con este análisis un tiempo aproximado de cuanto se duró procesando y dará resultados de cada jugadores por equipos también.

#### 5.2.7 Requirement language criteria

Los criterios de lenguaje descritos en este punto del documento ISO/IEC/IEEE 20148-2011 se están aplicando para el desarrollo de este documento.



### 5.2.8 Requirements attributes

Cargar un video digital almacenado en la dirección provista por el usuario../ Medio/ Prioridad: Alta

Descargar el video con los blobs y las etiquetas por equipo marcados. /Medio /Prioridad: Alta

Generar un informe en formato csv de la cantidad de jugadores de cada equipo, por cuadro. /Fácil /Prioridad:Media

Visualizar el número de jugadores detectados en cada cuadro, conforme transcurre el video./Difícil /Prioridad:Media

Visualizar el vídeo, con las regiones correspondientes a los jugadores o «blobs» visibles./Difícil /Prioridad:Alta

Visualizar las etiquetas por equipo (2 etiquetas) en los «blobs» correspondientes a los jugadores, a lo largo del vídeo. / Difícil /Prioridad:Media

Visualizar el tiempo en procesar el vídeo. /Facil /Prioridad:Media

Métrica de exactitud en los resultados: Cargar un archivo de ground truth con los blobs de los jugadores manualmente marcados, que permita además cuantificar la cantidad de fallos en la clasificación no supervisada./ Difícil/ Prioridad: Alta

Entre el procesamiento de blobs y el procesamiento del vídeo, habrá una dependencia ya que si existe un error en el proceso de carga del vídeo el resto del software no podrá procesar los blobs generados de dicho vídeo lo cual no permitirá obtener un resultado final.

nota: al final de cada requerimiento se indica la dificultad que tendrá poder desarrollarlo.

La seguridad es un factor fundamental en el desarrollo del proyecto ya que si otros equipos hackean el sistema tendrán todos los datos sobre el tipo de juego de Boca.

## 5.3 Practical considerations

### 5.3.1 Iteration and recursion of processes

#### 5.3.1.1 Iterative application of processes

Obtención de requerimientos, influye en el sistema. Análisis de los requerimientos, influye en el sistema. Diseño de Software, influye en el sistema. Programación del sistema, influye en el sistema. Aplicar métricas de calidad del sistema, influye en el sistema.

#### 5.3.1.2 Recursive application of processes

Obtención de requerimientos, influye en el conocimiento del sistema el cual influye en el sistema. Análisis de los requerimientos, influye en el conocimiento del sistema el cual influye en el sistema. Presentación de userstories, influye en el conocimiento del sistema el cual influye en el sistema. Presentación de casos de uso, influye en el conocimiento del sistema el cual influye en el sistema. Presentación de storyboards, influye en el conocimiento del sistema el cual influye en el sistema. Presentación de un prototipo, influye en el conocimiento del sistema el cual influye en el sistema. Luego de desarrollar cada funcionalidad se debe aplicar parámetros de calidad del software lo cual influye en el desarrollo del mismo y por ende en el resultado final del sistema.

## 5.4 Requirement information items

Ambiente de organización : Las personas de la junta directiva del Club Atlético Boca Juniors tienen como su principal objetivo obtener el título de campeón tanto de la liga de fútbol de Argentina como de la copa Libertadores de América.

Organización de negocios : Con el fin de obtener los títulos de campeones de la máxima competencia nacional (Argentina) y continental (Copa Libertadores de América) los dirigentes de Boca pretenden mejorar su juego tácticamente por medio de la implementación de un software el cual será desarrollado por las empresas Sport Analytics.

Operación del sistema: El sistema que desarrollará la empresa Sport Analytics contará con un procesamiento de video del cual se obtendrán datos del funcionamiento del equipo línea por línea (defensa, medio campo y delantera).

## 6 Processes

### 6.1 Requirement processes

Stakeholder: Directiva del Club Atlético Boca Juniors.

Requirements: El software cargará vídeos de fútbol que sean de formato mp4. El software analizará dichos vídeos en busca de blops para descubrir el parado del equipo línea por línea en dichos vídeos. Se generará un informe con las duraciones de las escenas procesadas, el informe irá en una tabla con información del equipo línea por línea y se dará en el idioma español. Se generará un informe con las posiciones de cada jugador en un cuadro, de un equipo específico, se dará el informe en una tabla donde por cada jugador se brinde la información obtenida en el informe y se dará en el idioma español. El proyecto contará con una interfaz de usuario, la cual permitirá a los dirigentes de CABJ “comunicarse” con el sistema.

#### 6.1.1 Guidelines for Processes

En este punto se hablará de toda la línea del proceso que tendrá o ha tenido el presente proyecto.

- 1) Stakeholder Requirements Definition Process: En este rubro se abarcan todas las reuniones que se tuvieron con la junta directiva del CABJ en donde estas personas especificaron sus deseos de contador con un software para mejorar la táctica de su equipo línea por línea.
- 2) Requirements Analysis Process: Esta característica de la línea del proceso, nos ayuda a darnos cuentas de cuáles de los requerimientos son realmente necesarios para la realización del mismo, los que se pueden hacer durante el proyecto y los que son irrealizables.
- 3) The architectural design process: Es el proceso donde se procede a hacer el diseño del software de acuerdo a los requerimientos acordados durante el análisis de requerimientos.
- 4) Developed process: Es el proceso en el cual el software se va a crear, en el mismo se intentan cumplir al pie de la letra el diseño antes establecido y por ende los requerimientos de usuario.
- 5) Quality process: Es el proceso en el cual se asegura la calidad del software, si bien es cierto se debe de implementar un sistema de calidad en cada una de las etapas de desarrollo se deben hacer testing de calidad para asegurar que el proceso que se ha venido desarrollando es un proceso de calidad.

6)Release process: Es el punto final del proyecto cuando se esta seguro de que el mismo funciona a la perfección, se hace su entrega y se culmina con el mismo.

7)Maintenance process: Se definirá con la junta directiva de boca si la empresa Sport Analytcs será la encargada de dar mantenimiento a este producto, de ser así, cada cuanto se hará.

## 6.2 Stakeholder requirements definition process

### 6.2.1 Purpose

En esta subsección se definen, como anteriormente se mencionó, todas la reuniones que se tienen con la junta directiva de CABJ, durante estar reuniones se definieron una serie de requerimientos que luego se analizaron mejor por los encargados de la producción del sistema para acordar con los directivos la entrega de los requerimientos definidos en la sección de requerimientos funcionales y no funcionales del sistema descritos en las subsecciones 9.4.3.3 hasta la 9.4.18

### 6.2.2 Outcomes

El sistema requerido es un analizador de vídeos utilizado para detectar jugadores por cuadros y observar su parado tactico en un terreno de juego para intentar mejorar linea por linea el funcionamiento del equipo.

### 6.2.3 Activities and tasks

Para completar el desarrollo de estos requerimientos definidos por el usuario, se emplearan las siguientes tareas:

Algoritmo de segmentación, Detección de las regiones candidatas de jugadores, Aprendizaje no supervisado, Creación de la interfaz gráfica, Conección entre interfaz y datos del sistema, Carga y Descarga de vídeos, Método para crear archivos en formato CSV, Método para calcular el tiempos de procesamientos del video.

## 6.3 Requirements analysis process

### 6.3.1 Purpose

Se deben analizar los requerimientos del usuario para saber cuales se pueden realmente abarcar durante el desarrollo del mismo, además entregarle al usuario métricas acerca de cuanto tiempo se durará para culminar con el proyecto y el precio de mismo.

### 6.3.2 Outcomes

Lo acordado después de este análisis se representó en los requerimientos antes mencionados.

### 6.3.3 Activities and tasks

Las actividades para este análisis fueron usar herramientas como Trello y Trello para organizar bien el proyecto y saber bien el tiempo que se llevará a cabo con el mismo.

## 6.4 Requirements engineering activities in other technical processes

### 6.4.1 Requirements in architectural design

En esta sección se crearan diseños para la implementación del proyecto, estos diseños serán en formato de diagrama de clases y de diagrama de componentes.

#### 6.4.1.1 Define the architecture y 6.4.1.2 Analyze and evaluate the architecture

No aplica ya que la arquitectura dada por el CABJ es la que se utilizará, así mismo las documentaciones necesarias y estudios o análisis para trabajar sobre la arquitectura del sistema se aplicaran a brindar por boca.

### 6.4.2 Requirements in verification

El proceso de verificación implica revisar arduamente que todos los requerimientos antes establecidos sean los que estén en el funcionamiento en el sistema.

### 6.4.3 Requirements in validation

El proceso de validación implica revisar arduamente que todos los requerimientos antes establecidos estén en correcto funcionamiento en el sistema.

## 6.5 Requirements management

### 6.5.1 Management Overview

Para el desarrollo de este proyecto es necesario contar con un correcto manejo de los requerimientos, para esto se creará un control o una manera de manejar que se cumpla correctamente el proceso establecido para la realización del proyecto

Para que esto sea posible, se utilizarán algunos estándares, para el documento se utilizará un estandar ISO/IEC/IEEE29148-201 para el documento de SyRS, además se cuenta con un estandar de codificación presente en la siguiente parte(Parte II) de este documento.

#### **6.5.2 Change management**

No Aplica, aún no hay cambios en los requerimientos por lo cual no se han hecho cambios en el manejo del proyecto

#### **6.5.3 Measurement for requirements**

##### **6.5.3.1 Plan measurement**

Las necesidades del cliente son los requerimientos antes establecidos, la mayor prioridad del sistema es procesar el video y permitir descargarlo con la información solicitada para que se pueda observar el parado tactico del equipo.

## **7 Information items**

Los documentos desarrollados para la informacion tanto de el grupo de trabajo como para el cliente son los siguientes.

Stakeholder requirements specification document (StRS)

System requirements specification document (SyRS)

Stakeholder requirements specification document

## **8 Guidelines for information items**

### **8.1 Requirements information item outlines**

En este caso la información esta brindada en el SyRS

### **8.2 Stakeholder requirements specification document**

No Aplica ya que en este documento solo se plantea el desarrollo del SyRS

### **8.3 System requirements specification document**

Se plantea a continuacion en la seccion 9.4

## **8.4 Software requirements specification document**

No Aplica ya que en este documento solo se plantea el desarrollo del SyRS

## **9 Information item content**

### **9.1 Introduction**

En este apartado se procede al desarrollo del SyRS, en el cual se especificarán datos generales del proyecto como el scope y título

### **9.2 General content**

Los contenidos generados se presentarn en el punto numero 2 del presente documento.

### **9.3 Stakeholder requirements specification (StRS) document.**

No Aplica ya que en este documento solo se plantea el desarrollo del SyRS

### **9.4 System requirements specification (SyRS) document**

Es el documento que se propone a desarrollar a continuación.

#### **9.4.1 System purpose**

El proposito de desarrollo del proyecto es generar un vídeo con analisis aplicados para tener una idea de las tacticas que esta utilizando el CABJ en sus partidos y con ello poder mejorar para la obtención de un nuevo campeonato para sus vitrinas.

#### **9.4.2 System scope**

Ver sección 1 de este documento.

#### **9.4.3 System overview**

##### **9.4.3.1 System context**

El sistema cuenta con una inerfaz de usuario, esta interfaz es web, en ella se permitirá cargar un video para su procesamiento, descargar el video procesado, generar informes de los resultados en formato csv, visualizaciones del video con diferentes datos especificados en los requerimientos del software, visualizar el tiempo del proceso del video.

Este sistema será utilizado por la junta directiva del CABJ y su director técnico, con el fin de mejorar el funcionamiento táctico del equipo.

Adjunto a este documento se encuentran diagramas de clases y de componentes que pueden ayudar al entendimiento del sistema.

#### 9.4.3.2 System functions

Las funciones que tendrá este sistema están descritas en la sección 9.4.4

#### 9.4.3.3 User characteristics

Los usuarios son una población muy reducida, ya que son únicamente los directivos y cuerpo técnico del Club Atlético Boca Juniors, el sistema se podrá acceder desde cualquier dispositivo ya que es web y se utilizará para intentar mejorar la parte táctica del equipo.

#### 9.4.4 Functional requirements

1. Cargar un video digital almacenado en la dirección provista por el usuario. Se recibirá un video que el usuario elija para ser cargado, una vez hecho esto se enviarán a los para crear blobs a ser procesados.
2. Descargar el video con los blobs y las etiquetas por equipo marcados. Tomará un video ya procesado y se lo brindará al usuario en el formato especificado en este documento.
3. Generar un informe en formato csv de la cantidad de jugadores de cada equipo, por cuadro. Se tomarán los datos generados durante el proceso (de cantidad de jugadores por cada equipo), con ellos se genera el informe en formato CSV.
4. Visualizar el número de jugadores detectados en cada cuadro, conforme transcurre el video. Se toma el video recibido, se le aplican los algoritmos de detección de jugadores dando como resultado un video en el cual se conoce el número de jugadores detectados en cada cuadro.
5. Visualizar el video, con las regiones correspondientes a los jugadores o «blobs» visibles. Se toma el video recibido, se le aplican los algoritmos de detección de jugadores dando como resultado un video en el cual se reconocen las regiones correspondientes a los jugadores visibles.



6. Visualizar las etiquetas por equipo (2 etiquetas) en los «blobs» correspondientes a los jugadores, a lo largo del vídeo. Se toma el video recibido, se le aplican los algoritmos de detección de jugadores dando como resultado un video en el cual se reconocen las etiquetas por equipo correspondientes a los jugadores .

7. Visualizar el tiempo en procesar el vídeo. No recibe nada, pero devuelve una variable modificada cada segundo para visualizar el tiempo de procesamiento del video.

8. Métrica de exactitud en los resultados: Cargar un archivo de ground truth con los blobs de los jugadores manualmente marcados, que permita además cuantificar la cantidad de fallos en la clasificación no supervisada. Recibirá un archivo de ground truth con los blobs y devolverá la cantidad de fallos en la clasificación no supervisada.

#### 9.4.5 Usability requirements

Debe tener ayudas para el usuario

Debe contar con un tutorial rápido para su uso adecuado

Debe contar con novedades para destacar sobre las otras aplicaciones deportivas de procesamiento de videos

Las métricas utilizadas para esta característica serán:

El tiempo que duren los usuarios en encontrar lo deseado.

El tiempo que duren los usuarios en sentirse cómodos con el sistema.

El tiempo que tarden los usuarios en aprender a usar el sistema.

#### 9.4.6 Performance requirements

No aplica, esta subsección hace mención a las variables que pueden influir de una manera externa al producto, en este caso el software puede seguir funcionando sea cual sea la situación para encontrar el parado táctico del equipo línea por línea durante los últimos partidos.

#### 9.4.7 System interfaces

La interfaz debe contar con una opción de carga y descarga de videos, además con una opción de generar informes, debe permitir visualizar un video (que contará con los datos previamente citados), debe permitir visualizar el tiempo de procesamiento del video y debe permitir cargar datos de archivos de ground truth, debe contar con opción de cuantificar los fallos en la clasificación no supervisada, debe permitir visualizar etiquetas por equipo y debe permitir visualizar el número de jugadores detectados en cada cuadro.

#### 9.4.8 System Operations

##### 9.4.8.1 Human system integration requirements

No aplica, el sistema es automático

##### 9.4.8.2 Maintainability

Para la mantenibilidad del sistema se contará con un ingeniero en sistemas encargado de ir a dar mantenimiento al sistema una vez cada 3 meses o cuando el sistema lo amerite.

##### 9.4.8.3 Reliability

9.4.9 System modes and states If the system can exist in various modes or states define these and, as appropriate, use diagrams.

No aplica, esto debido a que nuestro sistema es para un uso demasiado específico, en un ambiente único.

#### 9.4.10 Physical characteristics

##### 9.4.10.1 Physical requirements

Un servidor a características a convenir por la empresa sport analytics.

##### 9.4.10.2 Adaptability requirements

El sistema puede expandir la capacidad de usuarios con la cual se cuentan en el registro del sistema

#### 9.4.11 Environmental conditions

No aplica, el software a construir no depende de factores ambientales para su uso o desarrollo y no se verá comprometido por factores ambientales.

#### 9.4.12 System security

El sistema cuenta con un log-in en el cual se encuentran los datos personales de los directivos y cuerpo tecnico del CABJ(puede variar si la junta directiva o cuerpo técnico lo hace), con ello se asegura que nadie ajeno robe información acerca de como juega el club y obtenga ventaja de esto.

Además se debe tener una aplicación de seguridad dentro del sistema para asegurarse de que no puede “hackearse” fácilmente y así asegurar un tratamiento interno adecuado de los datos y que estos no caigan en manos de terceros.

#### 9.4.13 Information management

Para el almacenamiento de datos del club se contará con un equipo de servidores provistos por CABJ, el almacenamiento máximo será la capacidad máxima de almacenamiento de dichos servidores, aunque esto se hace pensando en una expansión del producto ya que los datos que hay que almacenar realmente son pocos.

#### 9.4.14 Policies and regulations

El sistema debe ser completamente en español ya que es el idioma nacional de argentina, lugar donde juega el CABJ

#### 9.4.15 System life cycle sustainment

El ciclo de vida del producto va desde la asignación de este proyecto hasta el momento en que es concluido, este ciclo de procesos se especifica más a fondo en la sección 6.1.1 de este documento, despues de cada sprint acordado por los desarrolladores del proyecto, se debe de revisar el ciclo de vida para ver si por algún motivo este se adelantó o atrasó.

#### 9.4.16 Packaging, handling, shipping and transportation

No aplica, este proyecto de software puede ser accedido desde internet, por lo mismo lo único que debe ser considerado en cuenta para este punto son los servidores del CABJ pero estos servidores están fuera de nuestra responsabilidad.

#### 9.4.17 Verification

La empresa sport analytics se compromete a verificar y validar las funcionalidades del sistema antes de entregarlo para así asegurarse que los requerimientos del sistema sean los establecidos por el usuario y que estén correctamente implementados en el software.

#### 9.4.18 Assumptions and dependencies

Para el correcto funcionamiento de este software se debe tener conexión a internet

## Part II

# Estandar de codicacion a utilizar a lo largo del proyecto semestral

## 1 Requerimientos

### 2.1 Nomenclatura

En esta seccion definiremos la forma de dar sentido al nombre de clases, variables, constantes, entre otros. Todo estara escrito en ingles incluyendo comentarios.

#### 2.1.1 Paquetes

Los paquetes se escribirán en minúsculas y sin utilizar caracteres especiales.

#### 2.1.2 Interfaces

Los nombres de interfaces utilizarán el prefijo Interface y estarán compuestos por palabras con la primera letra en mayúscula. Deben ser mezclas de mayúsculas y minúsculas, con la primera letra de cada palabra interna en mayúsculas. Se debe evitar el uso de abreviaciones que dificulten la comprensión del código y el objetivo de la interfaz. Por ejemplo: InterfaceServerConnection

**2.1.3 Clases** Los nombres de clases deben ser mezclas de mayusculas y minusculas, con la primera letra de cada palabra interna en mayusculas. Debemos intentar mantener los nombres de clases simples y descriptivos. Por ejemplo: `FigureCircle`

**2.1.4 Metodos** Los metodos deberan ser verbos (en innitivo), en mayusculas y minusculas con la primera letra del nombre en minusculas, y con la primera letra de cada palabra interna en mayusculas. No se permiten caracteres especiales. El nombre ha de ser lo sucientemente descriptivo. Por ejemplo: `calcPerimeter()`

**2.1.5 Variables** Los nombres de las variables tanto de instancia como estaticas reciben el mismo tratamiento que para los metodos. Las variables deberan en mayusculas y minusculas con la primera letra del nombre en minusculas, y con la primera letra de cada palabra interna en mayusculas. No se permiten caracteres especiales. El nombre ha de ser lo sucientemente descriptivo. Por ejemplo: `redCircle`

**2.1.6 Constantes** Los nombres de constantes deben escribirse todas en mayusculas con las palabras separadas por underscore. Todas seran declaradas como `private static nal`. Por ejemplo: `private static nal RED_CIRCLE`

## 2.2 Codicacion

### 2.2.1 Comentarios

Los comentarios seran utilizados para dar informacion adicional al desarrollador/ es sobre la implementacion del dise~no de la clase, metodo, interfaz, parametro, entre otros. Se tiene, por lo tanto, que evitar referencias al dise~no funcional de la misma. El uso abusivo de los comentarios es desaconsejable, principalmente por el trabajo extra necesario para su correcto mantenimiento. Se tienen que evitar el uso de caracteres especiales dentro de los comentarios.

### 2.2.2 Declaraciones

Para la declaracion de las variables se utiliza una declaracion de cada vez y no se permiten dejar variables locales sin inicializar. Por ejemplo: `private Circle = new Circle();`

## 2.3 Motivo de eleccion del estandar propuesto

El estandar propuesto ha sido elegido para mantener

### 2.3.1 Sentencias - Una sentencia por linea de codigo. -

Todo bloque de sentencias entre llaves, aunque sea una sola sentencia despu es de un `if`. Por ejemplo: `for(int salary = 0; salary < 3900000;`

salary++)

## 2.4 Notas.

### Buenas practicas de programacion

#### 2.4.1 Constantes

Como norma general todas las constantes numericas no deberan codificarse directamente, salvo la excepcion de -1, 0 y 1.

#### 2.4.2 Propiedades

El acceso/modificacion de las propiedades de una clase (no constantes) siempre mediante metodos de acceso get/set. La asignacion de variables / propiedades no podra ser consecutiva. salary1 = salary2 = "1000000" [No valido] No utilizar el operador asignacion en sitios donde se pueda confundir con el operador igualdad. Ni dentro de expresiones complejas.

#### 2.4.3 Metodos

No se debe acceder a un metodo estatico desde una instancia de una clase, debemos utilizar la clase en si misma.

## 2.5 Motivo por el que seleccionamos este estandar

La decision de implementar este estandar de codicacion, mas alla de contar como parte de la evaluacion del avance es una excelente forma de garantizar la futura mantenibilidad del codigo, ya sea por parte de nosotros mismos o diferentes grupos de trabajo. Esto porque existen mediciones/estudios que dicen que un 80% de la vida util de una pieza de software es necesario realizarle algun tipo de mantenimiento. Tambien el estandar le ofrece al codigo la posibilidad de tener una mejor legibilidad y mejor comprension. Dicho estandar propuesto es muy comun verlo en la mayoria de aplicaciones desarrolladas en Java, C y JavaScript, siendo esta una de las principales razones que nos llevo a elegirlo.

## 2.6 Herramienta para verificar el estandar de codicacion propuesto

Para la comprobacion del estandar propuesto se utilizara la herramienta llamada checkstyle. Checkstyle automatiza el proceso de verificacion de codigo en Java evitando el realizar esta tarea manualmente, la cual podria ser cansada y tediosa. Es altamente configurable y esta hecha para soportar casi todo estandar de codicacion.

### **3 Diagrama de componentes y digramas UML para la primera iteracion de la arquitectura del sistema**

#### **3.1 Diagrama de componentes**

#### **3.2 Diagramas UML**

### **4 Actividades de Aseguramiento de la calidad a realizar, una vez nalizado cada sprint**

El standar de IEE 730-2002 basicamente nos recomienda elaborar un plan para el Aseguramiento de la Calidad del Software.

#### 4.0.1 Tareas 4.0.2 Crear plan SQA

##### 4.0.2.1 Desarrollar y documentar el plan de SQA 4.0.2.2 Aprobar plan SQA

#### 4.0.3 Evaluar los requerimientos 4.0.3.1 Analizar requerimientos

##### 4.0.3.2 Proponer cambios a los requerimientos, si se considera necesario 4.0.3.3 Proponer posibles soluciones/escenarios para resolver los requerimientos

#### 4.0.4 Evaluar herramientas utilizadas

##### 4.0.4.1 Evaluar heramientas. Vericar resultados generados.

##### 4.0.4.2 Proponer nuevas herramientas o seguir utilizando las mismas

#### 4.0.5 Unit Testing 4.0.5.1 Mejoras en test unitarios

##### 4.0.5.1 Realizar nuevos test para las nuevas funcionalidades

#### 4.0.6 Evaluar la Administracion de la conguracion

##### 4.0.6.1 Congurar herramientas/ambientes involucradas en el sistema

#### 4.0.7 Resolucion de errores

##### 4.0.7.1 Identicar problema

##### 4.0.7.2 Analizar soluciones

##### 4.0.7.3 Vericar impacto de las soluciones propuestas

#### 4 4.0.8 Matriz de de responsabilidades

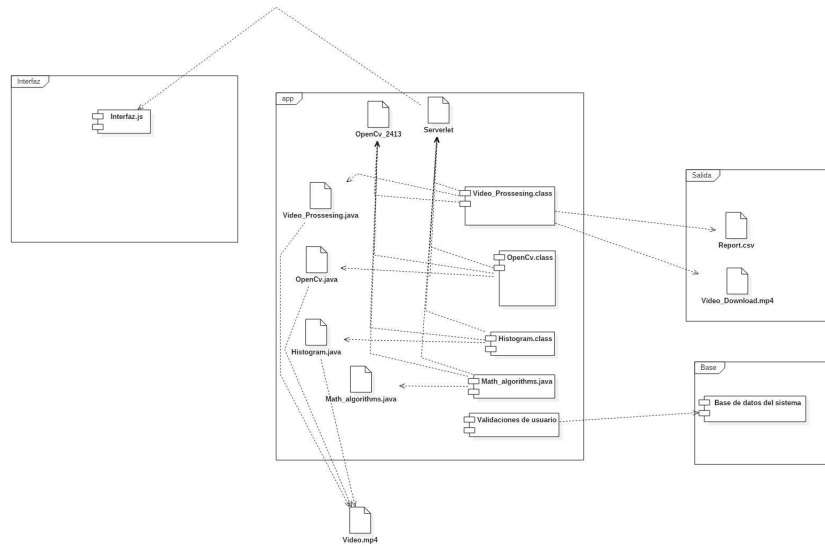
Tarea	Jean Carlo	Andres	Ricardo
4.0.2.1		x	
4.0.2.2	x	x	x
4.0.3.1	x	x	
4.0.3.2	x	x	
4.0.3.3	x	x	x
4.0.4.1	x		
4.0.4.2			x
4.0.5.1	x		
4.0.5.1	x		
4.0.6.1	x		x
4.0.7.1		x	x
4.0.7.2	x		x
4.0.7.3	x	x	x



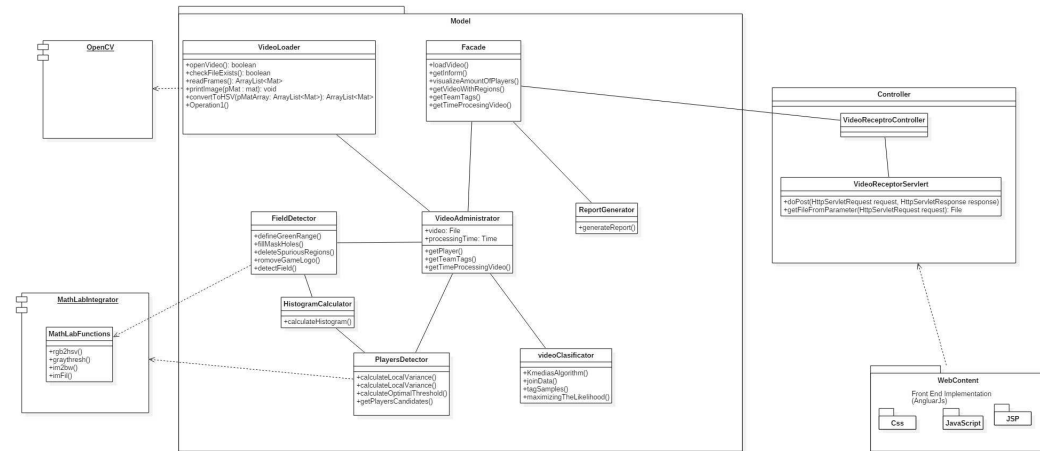
## Part III

# Diagramas

## Diagrama de componentes



## Diagrama de clases



Se eligió como patrón de diseño el utilizar el Facade. Este patrón busca simplificar el sistema, desde el punto de vista del cliente, proporcionando una interfaz unificada para un conjunto de subsistemas, definiendo una interfaz de nivel más alto. Facade va justo a la necesidad de simplificar la estructura compleja del proyecto y además al implementar este patrón obtenemos los siguientes beneficios:

- Oculta a los clientes de la complejidad del subsistema y lo hace fácil de usar.
- Favorece un acoplamiento débil entre el subsistema y sus clientes, consiguiendo que los cambios de las clases del sistema sean transparentes a los clientes.
- Facilita la división en capas y reduce dependencias de compilación.
- No se impide el acceso a las clases del sistema.

## Métricas:

1)

Funcionality	Characteristic	Metric	Required level	Tools
	Accutacy	Si la informacion generada es la correcta.	H	Junit
	Security	Que tan factible es la filtración de información a personas ajenas al equipo, ya que, si esta información cae en las manos de los otros equipos, serán armas para ganar los partidos lo que se traduce a perdidas económicas.	H	Fortify
	Interoperability	En cuantas plataformas va a funcionar el programa	M	Soapui

**Junit:** es una librería de java y se usa para ver si cada una de las funcionalidades estas empleadas de manera correcta, esto ayuda a medir la precisión para saber si la información generada es la correcta o no.

**Fortify;** es una herramienta usada para medir la seguridad y nos parece una buena opción para hacer el testing de Security

**Soapui:** se podría utilizar, si le mandamos una señal con las personas que logran ingresar al sistema desde diferentes navegadores o sistemas operativos para medir la interoperabilidad 2)

Reliability	Characteristic	Metric	Requited level	Tools
	Recoverability	Es importante tener una recuperación de la información si en algún momento se pierde, esto para ver un patrón en el cómo juegan al fútbol las diferentes líneas del equipo.	M	Metric 3
	Matutity	Ver cuantas pulgas tiene el software e intentar minimizarlas	H	Sonarqube
	Fault tolerance	Lo ponemos en baja prioridad ya que el sistema está funcionando para ver resultados a largo plazo, si falla, intentar arreglarlo es importante pero no es necesario que se arregle en el menor tiempo posible sino, que se puede hacer un trabajo con calma.	L	Sonarqube

**Metric 3:** Esta herramienta se utiliza cuando algún testing es difícil de aplicar, esto lo logra cargando datos similares para sacar una estadística.

**Sonarqube:** creemos que es la mejor opción para tratar estos 2 atributos de Reliability ya que, al detectar olores del código se hace más fácil una tener una buena práctica de programación, lo cual se pueden

encontrar más fácilmente las pulgas y corregirlas y se puede tener un mejor manejo de errores. 3)

Usability	Characteristic	Metric	Requited level	Tools
	Learnability	Que tan rápido los usuarios de nuestro software aprenderán a sacarle todo el provecho al mismo.	H	Click Heat
	Operability	Si nuestro software está cumpliendo con todos los requerimientos de los usuarios.	H	EclEmma
	Attractiveness	Entender cómo funciona el código, para un futuro mantenimiento.	M	Sonarqub

Click Heat es una buena herramienta si se puede tener una interacción con usuarios, esta herramienta permite saber, por medio de mediciones del calor, que cosas toca un usuario, a primera vista podría indicarnos si el usuario está intuyendo bien las funcionalidades de nuestro sistema.

EclEmma: esta herramienta nos permite saber cuáles partes del código son las que se están ejecutando, sabiendo esto, podremos saber si las que se ejecutan son las que cumplen con los requerimientos del usuario.

Sonarqub: nos permite tener un código más limpio y entendible para un futuro mantenimiento. 4)

Efficiency	Characteristic	Metric	Requited level	Tools
	Time behavior	El tiempo que tarda desde que se le da una instrucción hasta que se entrega un resultado.	M	JWatcher
	Resouerse	La totalidad de herramientas utilizadas para desarrollar el proyecto.	M	Reqview
	Utilization	La correcta ulización de los recursos para la realización del proyecto.	H	Reqview

**JWatcher:** es una herramienta de java utilizada para medir tiempos de comportamientos de los sistemas.

**Requview:** esta herramienta es seleccionada para estos atributos ya que, si manejamos de una manera optima los requerimientos del usuario, es más probable que logremos un correcto uso de los recursos y una correcta utilización del mismo.

5)

Maintainability	Characteristic	Metric	Requited level	Tools
	Stability	La cantidad de veces que el software da el resultado deseado.	H	Junit
	Changeability	La facilidad de hacerle modificaciones al código en un futuro.	M	Sonarqube
	Testability	Cantidad de test aplicados al proyecto.	M	Star UML

**Junit:** con esta herramienta podemos ver si el código que se está ejecutando.

**Sonrqube:** nos permite tener un código más limpio y entendible para una futura modificación.

**StarUML:** con esta herramienta podemos notar si tenemos un buen diseño de software, al saber esto, vamos a poder tener una noción de que tantas pruebas vamos a poder realizar al mismo 6)

Portability	Characteristic	Metric	Requited level	Tools
	Adaptability	El proyecto podría adaptarse a diferentes deportes como voleibol o baloncesto.	M	Sonarqube
	Installability	La facilidad de instalar el programa para ser utilizado.	M	Norton Ghost
	Replaceability	La facilidad de que un software en el mercado puede reemplazar el nuestro	L	Sonarqube

**Sonarqube:** con esta herramienta se puede identificar malas prácticas de diseño, como los olores del código, hecho esto se puede corregir y utilizar patrones de diseño que permitan la adaptación del código

**Norton Ghost:** para hacer imágenes de sistemas operativos donde “instalar” nuestro software y ver si realmente sirve la instalación correctamente.

**Sonarqube:** se elige esta herramienta ya que, si podemos identificar los errores en el código, será más sencillo entregar un trabajo cercano al 100% de calidad, lo cual, hará que sea más difícil que puede reemplazar a futuro.